

# An Edge Based Visual Tracking for Target within Complex Environment

En-Wei Huang<sup>1</sup>, Wei-Guan Yau<sup>1</sup> and Li-Chen Fu<sup>1,2</sup>

Department of Electrical Engineering<sup>1</sup>  
Department of Computer Science and Information Engineering<sup>2</sup>  
National Taiwan University, Taipei, Taiwan, R.O.C.

## Abstract

In this paper, a visual tracking system is proposed. The whole system is divided into two parts. The first part is the detection of the target, and the second part is the trajectory filtering. To detect the target in the image, we use the method based on the distribution of edges. In this method, the template is edge based, and the distribution of edges of the image is extracted, so that the two are convolved to find the location of the target. The method is more reliable than the traditional template matching in the complicated scene.

After finding the target in the image, we will control the camera platform to keep the target at the center of the image. The location of the target in the world coordinate frame is first transformed to the orientation relative to the camera whereby a tracking algorithm based on the IMM (Interacting Multiple Model) algorithm is developed. The algorithm estimates the angular velocity and the angular acceleration of the target relative to the camera, which along with some error information in the images are used for control of the camera platform. Finally, a number of experiments are conducted to validate the developed visual tracker.

**Keywords** : Visual tracking, target detection, distribution of edges, IMM algorithm.

## 1. Introduction

The aim of this paper is to design a monocular visual tracking system, which in principle is composed of a camera, a camera platform, and the processing unit. The camera can acquire the image including the object we are tracking. Through use of appropriate image processing technique, the object in the image can be precisely located, whereby the offset between the object and the image center can be found to calculate the control command of the camera platform. Since image taking and image processing normally take long time, the aforementioned control command synthesis will result in tracking delay. To

suppress such delay, it is necessary to predict the trajectory of the object so as to keep the object at the center of the image as much as possible.

In our experiments, the object to be tracked lies in a quite complicated background to test the ability of the developed detection method. Due to our hardware limitation, the angular velocity of the object relative to the camera platform is limited to 20 degree/sec to prevent from images blurring due to fast camera movement during tracking. Similarly, the limit on the angular acceleration is set at 30 degree/sec<sup>2</sup> because of the sampling rate and the finite view angle of the camera. The paper is organized as follows.

In section 2, the method of detecting the target in the image based on edge distribution is developed. In section 3, the tracking control of the camera platform based on the IMM (Interacting Multiple Model) algorithm[2-5] is proposed. To validate the developed visual tracker, computer simulation and experimental results are provided in section 4. Finally, the paper is concluded in section 5.

## 2. Visual Target Detection

In a visual tracking system, the essential goal is to detect the target correctly through obtained images. Failure to achieve this goal will render the whole system invalid. In this section, we will present our matching method based on edge distribution.

### 2.1 Template Matching Based on Edge

#### Distribution

In fact, in the task of visual tracking, it is impossible to declare that a specific method can solve all problems that occur during the course of target detection. According to the concept of the DSNR, whether the matching process can succeed depends on the complexity of the background.

To track a specific target, the shape of the target often contains much useful information. In other words, the shape, or equivalent to the distribution of the edges, is usually a better attribute to differentiate

two objects than the distribution of the gray level. On the other hand, it is also important to locate the target accurately, and to match against the shape of the template will provide a sharp peak of matching value contrary to the broad peak produced by the SAD method[1]. Hence, here we propose a matching method based on the distribution of the edges.

To save the computation time, we use a quite simple method to acquire the edges as follows. Note that the edge gradient can be described as

$$edge(x,y) = \frac{|g(x,y) - g(x+1,y)| + |g(x,y) - g(x,y+1)|}{2}$$

where  $g(x,y)$  means the gray level at  $(x,y)$ , and  $edge(x,y)$  means the intensity of the edge at location  $(x,y)$ . The template and the image will be preprocessed by the former edge gradient algorithm. However, the intensity of the edge at a location will not be preserved. Instead, it will be binarized into two groups via the following rule:

if  $edge(x,y) > Th1$   
 $EdgeExist(x,y) = 1$   
 else  
 $EdgeExist(x,y) = -1$

where  $Th1$  is a threshold, and  $EdgeExist(x,y)$  represents that an edge exists at  $(x,y)$ . After the original edge images of the template and the scene object are transformed into the binary type, the higher  $Th1$  is, the fewer the edge stamps will exist. Next, the distribution of the edge stamps of the template will be convolved with that of the object image according to the following:

$$Conv(i,j) = \sum_{x=0}^{T_x-1} \sum_{y=0}^{T_y-1} (EdgeExist_t(x,y) \times EdgeExist_o(x+i,y+j))$$

where  $EdgeExist_t$  is the distribution of the edges of the template, and  $EdgeExist_o$  is that of the object image. Thus, the location  $(i_M, j_M)$  where the value  $Conv(i_M, j_M)$  is maximized represents the location where the target exists.

## 2.2 Implementation of the Matching Process

### Process

The matching process can be divided into three parts. They are (a) building template (b) template matching (c) template updating. The whole matching process is shown in Fig. 1.

## 3. Control of Camera Platform

In this section, we will develop a method based on IMM (Interacting Multiple Model) algorithm to generate control signals for the camera platform.

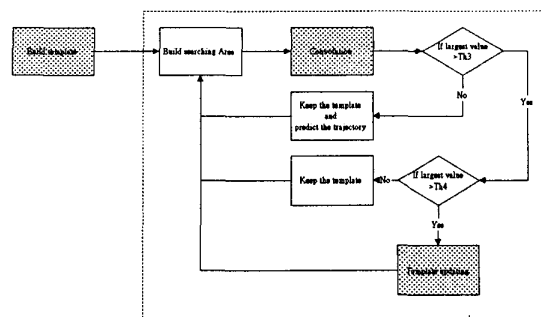


Fig. 1: The matching process

## 3.1 Design of Tracking Algorithm

In Fig. 2, we assume the target is projected to a semi-sphere with radius  $\hat{z}_c$ . Then we will drive the orientation of the camera,  $\phi$  and  $\theta$ , to track the orientation of the target,  $\alpha$  and  $\beta$ .

If we want to track the absolutely orientation of the target, we need to define the starting point of  $\phi$  and  $\theta$ . We also have to home the platform before processing. On the other hand, to track the absolutely orientation, we need very precise coordinate transform and agile platform, or we will suffer from the unstable system. For this reason, we design the algorithm tracking the angular velocity and acceleration of the target. It means that we want to match  $\dot{\phi}, \dot{\theta}$  and  $\ddot{\theta}$  with  $\dot{\alpha}, \dot{\beta}$  and  $\ddot{\beta}$  respectively.

To track a target, we have to keep the target inside the image on the first place. If we can track the angular velocity and acceleration of  $\phi$  and  $\theta$ , the target will be well kept in the image. According to Fig. 3, after tracking the angular velocity and acceleration, we then compensate the error of  $\phi$  and  $\theta$  by zeroing the offset values of  $u$  and  $v$  in the image, i.e.,

$$\Delta\phi = -\tan^{-1}(\Delta u \times dR_x)$$

$$\Delta\theta = -\tan^{-1}(\Delta v \times dR_y)$$

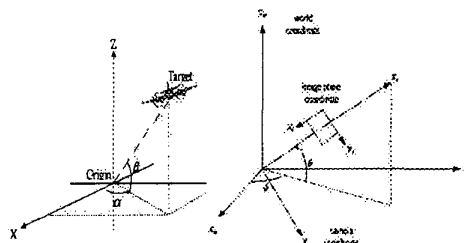


Fig. 2: Orientation of the target and camera relative to the world coordinate frame

where  $dR_x = \frac{d_x}{\lambda}$ ,  $dR_y = \frac{d_y}{\lambda}$ ,  $d_x$  is the pixel-width in the x direction,  $d_y$  is the pixel-length in the y direction and  $\lambda$  is the focal length of the camera. In turn, we can calculate the angular velocity and acceleration of the target with sample time  $T$  as

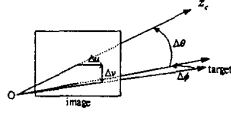


Fig. 3: Error in image,  $\phi$  and  $\theta$

$$\begin{aligned}\hat{\alpha}(k+1) &= \hat{\alpha}(k+1) + (\Delta\phi(k+1) - \Delta\phi(k))/T \\ \hat{\beta}(k+1) &= \hat{\beta}(k+1) + (\Delta\theta(k+1) - \Delta\theta(k))/T \\ \hat{\alpha}(k+1) &= (\hat{\alpha}(k+1) - \hat{\alpha}(k))/T \\ \hat{\beta}(k+1) &= (\hat{\beta}(k+1) - \hat{\beta}(k))/T\end{aligned}$$

By actuating the motors of the camera platform with the previous calculated motion commands, we should be able to keep the target at the center of the image. Thus the orientation of the target can be well tracked down in this way so that the disadvantages due to imprecise coordinate transform can be prevented as long as we compensate the error of  $\phi$  and  $\theta$  precisely.

For lack of the distance between target and the platform, we cannot depict the trajectory of the target in 3D space. However, a tracking task can be implemented as long as we have the orientation of the target. In our algorithm, we define the trajectory of the target in the space based on  $\alpha$  and  $\beta$ , which are independent with each other. On the other hand, because both  $(\phi, \theta)$  and  $(\Delta u, \Delta v)$  are independent with each other, we can track  $\alpha$  and  $\beta$  individually. It results in the simplicity of the motion models.

We define three kinds of models for the IMM algorithm. They are (a) constant angular velocity, (b) constant angular acceleration, (c) abrupt angular acceleration.

#### (a) Constant angular velocity(CAV)

The state transition equation of a linear system describing constant angular velocity is

$$\begin{bmatrix} \phi(n+1) \\ \dot{\phi}(n+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \phi(n) \\ \dot{\phi}(n) \end{bmatrix}$$

No matter what the value of the angular acceleration at step  $n$  is, it will be eliminated at the next step.

#### (b) Constant angular acceleration(CAA)

The state transition equation of a linear system describing constant angular acceleration is expressed by:

$$\begin{bmatrix} \phi(n+1) \\ \dot{\phi}(n+1) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \phi(n) \\ \dot{\phi}(n) \end{bmatrix}$$

In this kind of model, the acceleration at step  $n$  will be preserved at the next step, but the angular velocity will be updated depending on the time step  $T$  and the acceleration at step  $n$ .

#### (c) Abrupt angular acceleration(AAA)

The state transition equation of a linear system describing abrupt angular acceleration is

$$\begin{bmatrix} \phi(n+1) \\ \dot{\phi}(n+1) \end{bmatrix} = \begin{bmatrix} 1 & 2T \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \phi(n) \\ \dot{\phi}(n) \end{bmatrix}$$

In this case, we multiply the acceleration by two. Of course, the angular velocity will be updated in a way different from that with constant acceleration. The purpose of this model is to track a target with an abrupt motion.

Replacing  $(\phi, \dot{\phi})$  by  $(\theta, \dot{\theta})$ , we can get three identical equations for the vertical direction.

## 4. Simulations and Experiments

In this section, we first provide a simulation example to explain the effect of the IMM algorithm, and then provide an experimental result to demonstrate the performance of the developed visual tracking system.

### 4.1 Simulation

In our simulation, the cameras are both located at the origin of the world coordinate frame and suppose we will always capture the target if it is in the field of vision of the camera when it move in the 3D space. Through coordinate transform, we can simulate the process of a camera tracking the target.

In our simulation,  $dR_x = dR_y = 6 \times 10^{-4}$ , which will result in 22 degree horizontal view angle as and 16 degree vertical view angle. The initial conditions for the individual Kalman filters in IMM are

$$\mathbf{P}_{initial} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Where  $\mathbf{P}_{initial}$  is the state error covariance matrix  $\mathbf{P}$ ,  $\mathbf{Q}$  is the covariance matrix of the system noise  $\mathbf{W}$ ,  $\mathbf{R}$  is the covariance matrix of measurement noise  $\mathbf{V}$  and  $\mathbf{H}$  is the observation matrix.

In the IMM algorithm, we define the initial value of the likelihood as,  $[p_1 \ p_2 \ p_3]_{initial} = [0.34 \ 0.33 \ 0.33]$  and we define the switching probability matrix  $\mathbf{J}$  as

$$\mathbf{J} = \begin{bmatrix} 0.95 & 0.025 & 0.025 \\ 0.025 & 0.95 & 0.025 \\ 0.025 & 0.025 & 0.95 \end{bmatrix}$$

In our simulation, the sampling period is 100ms. The unit of distance is meter, and the unit of angle is radian. However, for clarity, we show the simulation result with degree.

In Fig. 4.1, we know that the target is located at (2,0,2) initially, and it has a circular motion with radius 1 m and period 10 seconds in counterclockwise direction. The camera platform has the initial orientation as  $\theta_{initial} = 0^\circ$  and  $\phi_{initial} = 40^\circ$ . We simulate the process for 100 steps, which means 10 seconds. Fig 4.1(a) shows the motion of the target.

Fig. 4.1(b) shows the trajectory in image. First, the target appears in the upper of the image. In first

three steps, we build the initial conditions for the IMM algorithm. Because in this stage the servo system will not operate, we can find the trajectory moves toward left. After building the initial conditions, we can find that the trajectory in image will concentrate on the center of the image. Fig. 4.1(c) shows the errors in image in both directions,  $u$  and  $v$ .

Fig. 4.1(d) shows the angular velocity of  $\dot{\phi}$  and  $\dot{\alpha}$ . We can find that  $\dot{\phi}$  seems not to track the  $\dot{\alpha}$  well at the beginning. It is because that we have a large error in  $u$  direction. To compensate the error in  $u$  direction, it will result in a  $\dot{\phi}$  different from  $\dot{\alpha}$ . Fig. 4.1(e) shows the error between  $\dot{\phi}$  and  $\dot{\alpha}$ .

The same phenomenon also appears in the angular velocity of  $\dot{\theta}$  and  $\dot{\beta}$ . To compensate the error in  $v$  direction, the angular velocity also will not match well enough at the beginning. Fig. 4.1(f) shows  $\dot{\theta}$  and  $\dot{\beta}$ , and Fig. 4.1(g) shows the error between  $\dot{\theta}$  and  $\dot{\beta}$ .

It is interesting to see the last two figures. Fig. 4.1(h) and Fig. 4.1(i) show the likelihood ratios of the three models for  $\dot{\phi}$  and  $\dot{\theta}$  respectively. In this case, because the target changes its velocity smoothly, the third model, the abrupt angular acceleration model, always keeps in a low likelihood ratio. On the other hand, because the target is in a circular motion, the trajectory projects on the  $z$  axis and  $y$  axis will be in simple harmonic motions. We can find that the likelihood ratios of the other two models, the constant angular velocity model and the constant angular acceleration model, match the motion of the target correctly.

## 4.2 Experiment

In this subsection, we will briefly introduce the equipment in our system. Our visual tracking system is composed of these three major parts:

- (a) camera
- (b) camera platform
- (c) processing unit

In our visual tracking system, we use a monochrome CCD camera with resolution of  $640 \times 480$ . The angle of view of this camera is  $20^\circ$  from the left to the right and  $16^\circ$  from the top to the bottom. The resolution is enough for the tracking task in general case. The camera platform is the servo system of our visual tracking system. The camera platform has two rotating axes. Each of these two axes is driven by a stepping motor, which has 10000 counts per revolution. Hence, the camera platform can provide a resolution of  $0.036^\circ$  in each axis. This function is provided by the ADVANTECH.

The processing unit consists of two elements, namely, a PC and a DSP based image processing card.

The image processing card has a TMS320C40 DSP on it, and the image is acquired and processed by this card. After the image has been processed, the trajectory of the camera platform is computed by the PC.

In our experiments, we use the moving hand as the tracking target. The unit of the angular velocity is degree/sec, and the unit of the angular acceleration is degree/sec<sup>2</sup>. The unit of the control command is degree/step.

Figure 4.2 shows the sequence that a hand moves slowly and shaken abruptly from a clear background into a complicated background, which is formed by an uncovered computer. Because of the noisy background, the peak of the result of convolution may not be very sharp. But there are no multiple peaks happened.

In Fig 4.3(d), the control command of axis 2 closely follows the trend of the estimated angular velocity, and we can find that in Fig. 4.3(f), the error in either direction is compensated right away. But since the hand accelerates and decelerates abruptly, overshoots are generated in the trajectory of the error (center-miss) in axis 2. However, the error converges normally in 15 steps.

## 5. Conclusions

In this paper, we have proposed a method of detecting the target with complicated background based on the edge distribution in a sequence of computer images. After discriminating the target from the background, the core of our tracking algorithm based on the IMM (Interacting Multiple Model) algorithm controls the camera platform to lock the target at the center of the image.

The developed method based on the distribution of image edges hears a number of advantages: (1) a sharp peak of the matched value, (2) insensitivity to the variance of light, and (3) higher immunity to additive noise. On the other hand, the tracking algorithm based on the IMM algorithm provides higher accuracy of prediction of the trajectory of a maneuvering target.

## References

- [1] R. Brunelli and T. Poggio, "Template Matching: Matched Filter and Beyond", *Pattern Recognition*, Vol. 30, No.5, pp.751-768, 1997.
- [2] E. Mazor, A. Averbuch, Y. Bar-Shalom and J. Dayan, "Interacting Multiple Models in Target Tracking: A Survey", *IEEE Trans. on Aero. and Electro. Sys.*, Vol. 34, No. 1, Jan 1998.
- [3] Y. Bar-Shalom, K. C. Chang and H. A. P. Blom, "Tracking a Maneuvering Target Using Input

Estimation Versus the Interacting Multiple Model Algorithm”, IEEE Trans. on Aero. and Electro. Sys., Vol. AES-25, No. 2, Mar 1989.

- [4] Kevin J. Bradshaw, Ian D. Reid and David W. Murry, “ The Active Recovery of 3D Motion Trajectories and Their Use in Prediction”, IEEE Trans. on Pattern Analysis and Mach. Intell., Vol. 19, No. 3, Mar 1997.
- [5] A. Munir and D. P. Atherton, “ Maneuvering Target Tracking Using Different Turn Rate Models in the Interacting Multiple Model Algorithm”, Proc. of the 34<sup>th</sup> Conf. on Decision and Control, Dec 1995.

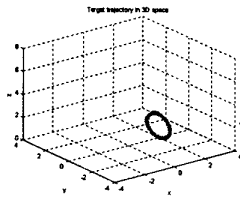


Fig. 4.1(a)

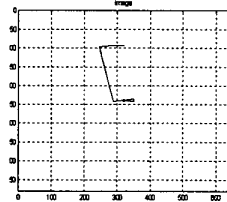


Fig. 4.1(b)

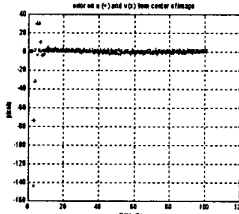


Fig. 4.1(c)

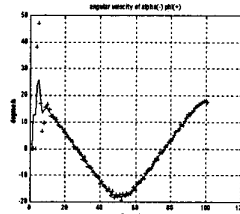


Fig. 4.1(d)

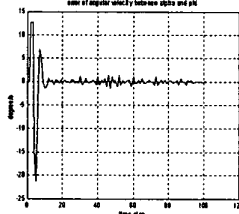


Fig. 4.1(e)

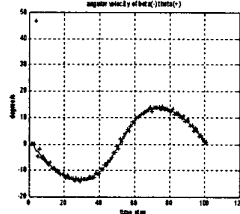


Fig. 4.1(f)

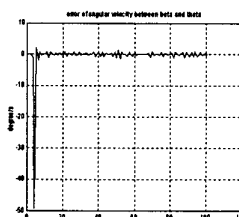


Fig. 4.1(g)

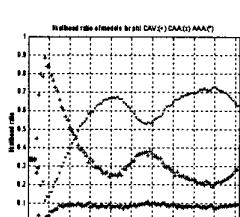


Fig. 4.1(h)

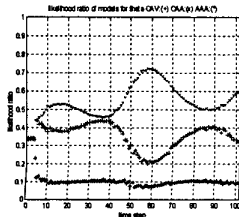


Fig. 4.1(i)

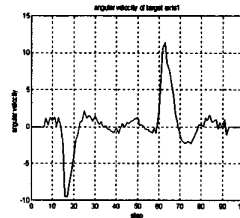


Fig. 4.3(a)

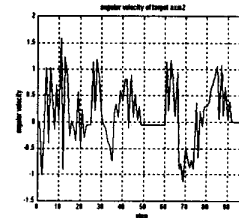


Fig. 4.3(b)

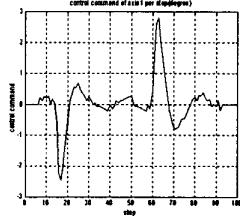


Fig. 4.3(c)

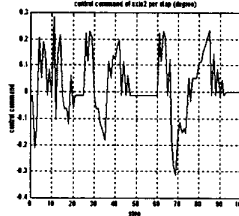


Fig. 4.3(d)

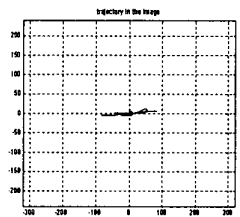


Fig. 4.3(e)

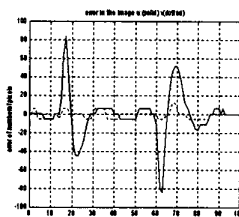


Fig. 4.3(f)

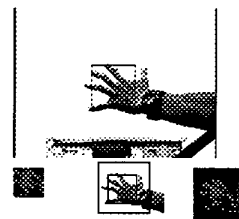


Fig. 4.2(a)

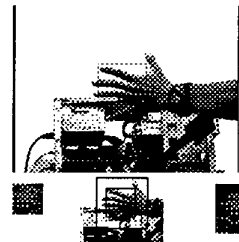
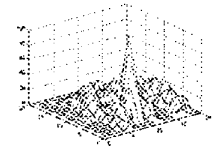


Fig. 4.2(b)

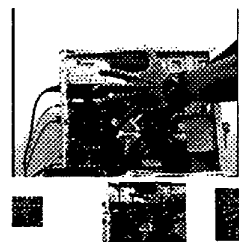
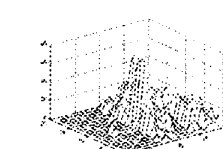


Fig. 4.2(c)

